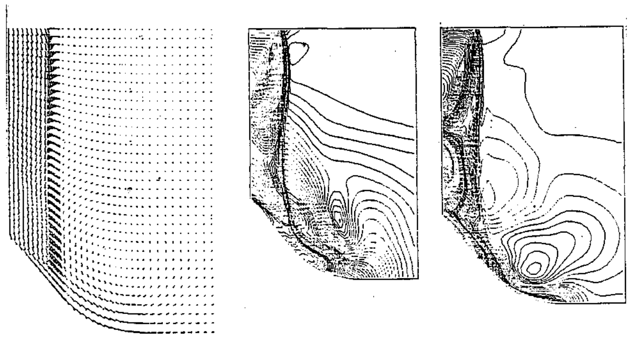

VECTOR DENSITY CONTOURS MACH CONTOURS

**Fig. 7 Vector, density and Mach contours for jet deflector: $M_e = 2.2$ and $p_e/p_a = 1.2$, $X_c/De = 3$.**

flow domain where the solution is smooth, but are "switch-off" in the region of shock waves. A term involving second differences is then "switch-on" to damp oscillations near the shock waves. This switching is achieved by means of shock wave sensor, based on the local second differences of pressure.

The numerical computations were done on Landmark i860 base work station with different grid sizes such as $150 \times 40$, $200 \times 40$ and $150 \times 50$.

## Results and Discussion

The schlieren picture of the free jets at $M_e = 2.2$ and $p_e/p_a = 1.2$ is depicted in Fig. 2a. The schlieren picture shows most of the flowfield features of the free jets, such as jet boundaries, expansion waves, jet shock, Mach disk, reflected shocks, etc. The impingement flowfield observed through the schlieren system is shown in Figs. 2b and 2c. The flowfield upstream of the cone shock is somewhat similar to that of the free jets as can be seen from the schlieren pictures. It can be seen from the schlieren photograph that the free jets Mach disk has been displaced upstream by about $0.13De$ due to presence of the deflector at $X_c/De = 2$. A weak compression region is observed close to deflector apex. Several compression and expansion zones are observed downstream of the deflector apex. The compression takes place due to the deflector curvature $R_2$ with an increase in distance, i.e., at $X_c/De = 3$, the jet deflector is now placed in the second shock cell of the free jets. The flowfield has changed appreciably as compared to the flowfield at $X_c/De = 2$. A stronger conical shock has appeared because the incoming jet flow has become supersonic before impinging on the deflector wall.

Figure 3 shows the pressure variation along the jet axis for free jets at $M_e = 2.2$ and $p_e/p_a = 1.2$ and 0.8. The comparison between the experimental and numerical results shows good agreement in pressure distributions along the centerline of the free jets. Figure 4 displays the vector, density, and Mach contours plots for the underexpanded free jet. The shock cells are also visible in the contour plots. Thus, the over all flowfield features of the supersonic free jets is very well captured by present numerical computations.

The static pressure distribution on the deflector surface is nondimensionalized by the settling chamber pressure $P_0$. The location of the static pressure port on the deflector surface is indicated as the distance $r$ measured from the model axis in the direction normal to it; $r$ is nondimensionalized by the nozzle exit diameter $De$. The measured static pressure distributions over the jet deflector surface are shown in Fig. 5.

The pressure distribution at $X_c/De = 2$ is shown in Fig. 5a. The stagnation point pressure has a value of 0.232. The static pressure $p/P_0$ increases marginally to a pressure of 0.305 at $r/De = 0.173$. Further, it starts to decrease to a minimum value of $p/P_0 = 0.0308$ at $r/De = 0.681$, which is below ambient pressure. The pressure further increases due to deflector curvature, attains a peak value of $p/P_0 = 0.1888$ at $r/De = 1.15$, and subsequently falls due to mixing with the ambient condition.

At $X_c/De = 3$, the stagnation point pressure has a higher value of $p/P_0 = 0.44$ as compared to $X_c/De = 2$. The pressure falls rapidly to $r/De = 0.14$ when a small pressure jump is observed, as seen in Fig. 5b. Farther downstream the pressure attains a minimum value of $p/P_0 = 0.04$ at $r/De = 0.55$. The pressure distribution beyond this point is similar to the behavior at $X_c/De = 2$. It can be seen from Fig. 5 that a comparison between experimental and numerical results shows good agreement.

Figures 6 and 7 depict vector, density, and Mach contour plots over the jet deflector. The contours plots exhibit all of the essential flowfield features like Mach disk, jet boundary, expansion and compression waves, cone shock, etc. The comparison of the schlieren pictures of the free jets and impinging jets reveals that the impingement flowfield has similar characteristics of the free jets up to the cone shock of the deflector's apex. The density and Mach contours of the free jets have somewhat similar flowfield features as compared with the density and Mach contours of the impinging flowfield. But the vector plots of the impinging free jets differ appreciably from the impinging vector plots. The pressure distributions above the deflector surface are useful for aerodynamic and structural design.

## References

[1]Seiner, J. M., and Norum, T. D., "Experiments of Shock Associated Noise on Supersonic Jets," AIAA Paper 79-1326, July 1979.

[2]Reid, J., and Hasting, R. C., "The Effect of a Central Jet on the Base Pressure of a Cylindrical After Body in a Supersonic Stream," Aeronautical Research Council, Report and Memorandum 3224, London, 1961.

[3]Chiang, C.-H., "Axially Symmetric Supersonic Turbulent Jets Discharged from a Nozzle with Underexpansion," *Turbulent Jets of Air, Plasma, and Real Gas*, Constants Bureau, New York, 1969, pp. 111–137.

[4]Chuech, S. G., Lia, M. C., and Faeth, G. M., "Structure of Turbulent Sonic Underexpanded Jets," *AIAA Journal*, Vol. 27, No. 5, 1989, pp. 549–559.

[5]Love, E. S., Grigsby, C. E., Lee, L. P., and Woodling, N. J., "Experimental and Theoretical Studies of Axisymmetric Free Jets," NASA TR R 6, 1959.

[6]Abel-Fattah, A. M., "Discrete Tone Emission from High Pressure Ratio Supersonic Jets from Convergent-Divergent Nozzles," *AIAA Journal*, Vol. 26, No. 3, 1988, pp. 283–291.

[7]Dash, S. M., and Wolf, D. F., "Fully Coupled Analysis of Jet Mixing Problem, Part 1, Shock Capturing Model, SCIPVIS," NASA CR 3761, Jan. 1984.

[8]Sinha, R., Zakky, V., and Erdos, J., "Flowfield Analysis of Plumes of Two Dimensional Underexpanded Jets by a Time Dependent Method," *AIAA Journal*, Vol. 9, No. 12, 1970, pp. 2363–2370.

[9]Dash, S. M., Pearce, B. E., and Pergament, H. S., "Prediction of Rocket Plume Flowfields for IR Signatures Studies," *Journal of Spacecraft and Rockets*, Vol. 17, No. 3, 1980, pp. 190–199.

[10]Lamont, P. J., and Hunt, B. L., "Impingement of Underexpanded Axisymmetric Jets on Perpendicular and Inclined Flat Plate," *Journal of Fluid Mechanics*, Vol. 100, Pt. 3, 1980, pp. 471–511.

[11]Jennions, I. K., and Hunt, B. L., "The Axisymmetric Impingement of Supersonic Air Jets on Cones," *Aeronautical Quarterly*, Vol. 31, Feb. 1980, pp. 26–41.

[12]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of Euler Equations by Finite Volume Method Using Runge-Kutta Time-Stepping Scheme," AIAA Paper 81-1259, June 1981.

# Solution of Keller's Box Equations for Direct and Inverse Boundary-Layer Problems

Tarek M. A. El-Mistikawy*
*Cairo University, Giza 12211, Egypt*

## Introduction

THE difference equations that result from applying Keller's box scheme to the direct boundary-layer problem have tradi-

Received May 17, 1993; revision Dec. 23, 1993; accepted for publication Dec. 30, 1993. 

*Lecturer, Department of Engineering Mathematics and Physics, Faculty of Engineering.

tionally been solved by casting the matrix of coefficients into block tridiagonal (BTD) form of $3 \times 3$ blocks. Then, use is made of the block tridiagonal matrix inversion algorithm (BTDMIA).[1]

When applied to the inverse boundary-layer problem, the BTD form is destroyed, since line unknowns such as the pressure introduce additional columns. Although the Mechul function method[1] of Cebeci and Keller makes it possible to recast the matrix into a BTD form, the blocks are $4 \times 4$, which translates into considerable increase in computational time and storage requirements. The bordering algorithm of Chen and Cebeci[2] maintains the $3 \times 3$ BTD form but requires three inversion processes. A more efficient application of the bordering technique is Williams' eigenvalue method,[3] which requires two inversion processes only on a $3 \times 3$ BTD matrix.

This Note presents a superior alternative to the BTDMIA as applied to both the direct and inverse boundary-layer problems.

## Description of the Problem

Let Keller's box scheme be applied on a grid of $J + 1$ points across the boundary layer to an inverse problem involving $n$ grid-point unknowns and $k$ line unknowns. Given are $m$ conditions at one side of the layer to be marked with $j = 0$ and $m'$ conditions at the other side that is to be marked with $j = J$ $(0 \le m' \le m \le n)$. In addition, $k'$ multipoint conditions ($k' \ge 0$, $m + m' + k' = n + k$) hold.

The resulting equations can be expressed in matrix form as

$$\Gamma z = g \tag{1}$$

The vector of unknowns $z$ is written as follows:

$$(x_0\ y_0\ x_1\ y_1 \dots x_{j-1}\ y_{j-1}\ x_j\ y_j \dots x_{J-1}\ y_{J-1}\ x_J\ y_J\ \ q)^T$$

Its subvectors $x_j$ and $y_j$ ($j = 0 \to J$) contain $m$ and $n-m$ of the $j$th grid-point unknowns, respectively, whereas its subvector $q$ contains the $k$ line unknowns. For more generality, $z$ and its components are enlarged to contain $\ell$ columns ($\ell \ge 1$) corresponding to a similar enlargement of the right-hand-side vector $g$.

The augmented matrix of coefficients $\Gamma_+ = [\Gamma \mid g]$ has the form

$$\left[\begin{array}{cccccccccc}
C_{20} & D_{20} & & & & & & & E_{20} & g_{20} \\
A_{11} & B_{11} & C_{11} & D_{11} & & & & & E_{11} & g_{11} \\
A_{21} & B_{21} & C_{21} & D_{21} & & & & & E_{21} & g_{21} \\
\hline
& & A_{1j} & B_{1j} & C_{1j} & D_{1j} & & & E_{1j} & g_{1j} \\
& & A_{2j} & B_{2j} & C_{2j} & D_{2j} & & & E_{2j} & g_{2j} \\
\hline
& & & & A_{1J} & B_{1J} & C_{1J} & D_{1J} & E_{1J} & g_{1J} \\
& & & & A_{2J} & B_{2J} & C_{2J} & D_{2J} & E_{2J} & g_{2J} \\
& & & & & & A_{3J+1} & B_{3J+1} & E_{3J+1} & g_{3J+1} \\
A_{41} & B'_{41} & A_{4j} & B'_{4j} & A_{4J} & B'_{4J} & A_{4J+1} & B'_{4J+1} & E'_4 & g'_4
\end{array}\right]$$

Respectively, the submatrices named $A$, $B$, $C$, $D$, $E$, and $g$ have $m$, $n$–$m$, $m$, $n$–$m$, $k$, and $\ell$ columns. The number of rows is indicated by the leading subscript 1, 2, 3, or 4 to be $n$–$m$, $m$, $m'$, or $k'$.

The first $m$ rows of $\Gamma_+$ correspond to the ($j = 0$) conditions

$$C_{20}x_0 + D_{20}y_0 + E_{20}q = g_{20} \tag{2}$$

which are arranged so that $C_{20}$ is nonsingular. It is this requirement that decides which grid-point unknowns are to be contained in the subvectors named $x$.

Each of the $J$ subsequent sets of $n$ rows corresponds to Keller's finite difference representation having the form

$$A_{1j}x_{j-1} + B_{1j}y_{j-1} + C_{1j}x_j + D_{1j}y_j + E_{1j}q = g_{1j} \tag{3a}$$

$$A_{2j}x_{j-1} + B_{2j}y_{j-1} + C_{2j}x_j + D_{2j}y_j + E_{2j}q = g_{2j} \tag{3b}$$

of the boundary-layer equations over the $j$th box, starting from the first that is adjacent to $j = 0$. The arrangement of the $n$ equations within this form [Eqs. (3a) and (3b)] is arbitrary, allowing for partial pivoting.

The next set of $m'$ rows corresponds to the ($j = J$) conditions

$$A_{3J+1}x_J + B_{3J+1}y_J + E_{3J+1}q = g_{3J+1} \tag{4a}$$

The last $k'$ rows correspond to the multipoint conditions

$$\sum_{j=0}^{J} A_{4j+1}x_j + \sum_{j=0}^{J} B'_{4j+1}y_j + E'_4\, q = g'_4 \tag{4b}$$

{As an illustration, consider the problem described by Eqs. (5–7) of Ref. 2. There are three grid-point unknowns, $\delta f$, $\delta u$, and $\delta v$, and one line unknown, $\delta w$; i.e., $n = 3$ and $k = 1$. Having two surface conditions, Eqs.[2] (6a) and (6b), and two far-field conditions, Eqs.[2] (6c) and (7), means that $m = m' = 2$ and $k' = 0$, and that $j = 0$ can be assigned to either the surface or a far-field position. Let the former alternative be adopted and the vectors $x_j = [\delta f_j\ \delta u_j]^T$, $y_j = [\delta v_j]$, and $q = [\delta w]$ be formed. Then, Eqs.[2] (6a) and (6b) give

$$C_{20} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{(nonsingular)}, \qquad D_{20} = E_{20} = g_{20} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

whereas Eqs.[2] (5c), (5a), and (5b) give, for $j = 1 \to J$, and with $\theta_j = h_{j-1}/2$,

$$A_{1j} = [(s_4)_j (s_6)_j], \quad B_{1j} = [(s_2)_j], \quad C_{1j} = [(s_3)_j (s_5)_j]$$

$$D_{1j} = [(s_6)_j], \quad E_{1j} = [(s_7)_j], \quad g_{1j} = [(r_2)_j]$$

$$A_{2j} = \begin{bmatrix} -1 & -\theta_j \\ 0 & -1 \end{bmatrix}, \quad B_{2j} = D_{2j} = \begin{bmatrix} 0 \\ -\theta_j \end{bmatrix}, \quad C_{2j} = \begin{bmatrix} 1 & -\theta_j \\ 0 & 1 \end{bmatrix}$$

$$E_{2j} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad g_{2j} = \begin{bmatrix} (r_1)_j \\ (r_3)_{j-1} \end{bmatrix}$$

and, finally, Eqs.[2] (6c) and (7) give

$$A_{3J+1} = \begin{bmatrix} 0 & 1 \\ \gamma_1 & 0 \end{bmatrix}, \quad B_{3J+1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad E_{3J+1} = \begin{bmatrix} -1 \\ \gamma_2 \end{bmatrix}, \quad g_{3J+1} = \begin{bmatrix} 0 \\ (r_3)_J \end{bmatrix}$$

with no $A_{4j}$, $B'_{4j}$, $E'_4$, or $g'_4$ to define.}

## Solution Procedure

We introduce recurrence relations of the form

$$y_{j-1} = \bar{y}_{j-1} - \tilde{Y}_{j-1}y_j - \hat{Y}_{j-1}q \qquad j = 1 \to J \tag{5a}$$

$$x_j = \bar{x}_j - \tilde{X}_j y_j - \hat{X}_j q \qquad j = 0 \to J \tag{5b}$$

where the $\bar{\ }$, $\tilde{\ }$, and $\hat{\ }$ components of $y_{j-1}$ and $x_j$ have $\ell$, $n$–$m$, and $k$ columns, respectively.

Comparison of the ($j = 0$) level of Eq. (5b) with Eq. (2) gives

$$C_{20}[\bar{x}_0\ \tilde{X}_0\ \hat{X}_0] = [g_{20}\ D_{20}\ E_{20}] \tag{6}$$

Use of the $(j-1)$ level of Eq. (5b) to eliminate $x_{j-1}$ from Eqs. (3a) and (3b) and then comparison with Eqs. (5a) and (5b) gives, for $j = 1 \to J$,

$$\begin{bmatrix} \tilde{B}_{1j} & C_{1j} \\ \tilde{B}_{2j} & C_{2j} \end{bmatrix} \begin{bmatrix} \bar{y}_{j-1} & \tilde{Y}_{j-1} & \hat{Y}_{j-1} \\ \bar{x}_j & \tilde{X}_j & \hat{X}_j \end{bmatrix} = \begin{bmatrix} \bar{g}_{1j} & D_{1j} & \hat{E}_{1j} \\ \bar{g}_{2j} & D_{2j} & \hat{E}_{2j} \end{bmatrix} \tag{7}$$

where, for $i = 1, 2$,

$$\bar{g}_{ij} = g_{ij} - A_{ij}\bar{x}_{j-1} \tag{8a}$$

$$\tilde{B}_{ij} = B_{ij} - A_{ij}\tilde{X}_{j-1} \tag{8b}$$

$$\hat{E}_{ij} = E_{ij} - A_{ij}\hat{X}_{j-1} \tag{8c}$$

Use of the $(j = J)$ level of Eq. (5b) to eliminate $x_J$ from Eq. (4a) and successive use of Eq. (5b) to eliminate $x_j$ and then Eq. (5a) to eliminate $y_j$ (for all possible $j = 0 \to J$) from Eq. (4b) give

$$\begin{bmatrix} \tilde{B}_{3J+1} & \hat{E}_{3J+1} \\ \tilde{B}_{4J+1} & \hat{E}_{4J+1} \end{bmatrix} \begin{bmatrix} y_J \\ q \end{bmatrix} \begin{matrix} = \\ = \end{matrix} \begin{bmatrix} \bar{g}_{3J+1} \\ \bar{g}_{4J+1} \end{bmatrix} \tag{9}$$

where $\bar{g}_{iJ+1}$, $\tilde{B}_{iJ+1}$, and $\hat{E}_{iJ+1}$ $(i = 3, 4)$ are given by Eqs. (8a–8c). For $i = 4$, this requires prior evaluation of $g_{4J+1}$, $B_{4J+1}$, and $E_{4J+1}$ from the following successive calculations:

$$\left.\begin{aligned} g_{41} &= g'_4,\ g_{4j+1} = \bar{g}_{4j} - \tilde{B}_{4j}\bar{y}_{j-1} \\ B_{41} &= B'_{41},\ B_{4j+1} = B'_{4j+1} - \tilde{B}_{4j}\tilde{Y}_{j-1} \\ E_{41} &= E'_4,\ E_{4j+1} = \hat{E}_{4j} - \tilde{B}_{4j}\hat{Y}_{j-1} \end{aligned}\right\} j = 1 \to J \tag{10}$$

where $\bar{g}_{4j}$, $\tilde{B}_{4j}$, and $\hat{E}_{4j}$ $(j = 1 \to J)$ are given by Eqs. (8a–8c).

The solution procedure can be implemented in two FORTRAN-DO-LOOPs. Its steps, now that all pertinent relations have been laid down, are enumerated as follows:

1) Solve Eq. (6) to get $\bar{x}_0$, $\tilde{X}_0$, and $\hat{X}_0$, and assign to $g_{4J+1}$, $B_{4J+1}$, and $E_{4J+1}$ their initial values given by the left members of Eqs. (10).

2) In the $j$th step $(j = 1 \to J)$ of the first LOOP: a) evaluate $\bar{g}_{ij}$, $\tilde{B}_{ij}$, and $\hat{E}_{ij}$, for $i = 1, 2, 4$ using Eqs. (8a–8c); b) solve Eq. (7) to get $\bar{y}_{j-1}$, $\tilde{Y}_{j-1}$, $\hat{Y}_{j-1}$, $\bar{x}_j$, $\tilde{X}_j$, and $\hat{X}_j$; and then c) update $g_{4J+1}$, $B_{4J+1}$, and $E_{4J+1}$ using the right members of Eqs. (10).

3) Evaluate $\bar{g}_{iJ+1}$, $\tilde{B}_{iJ+1}$, and $\hat{E}_{iJ+1}$ for $i = 3, 4$ using Eqs. (8a–8c) and then solve Eq. (9) to get $y_J$ and $q$.

4) In the $j$th step $(j = J \to 1)$ of the second LOOP, calculate $y_{j-1}$ and $x_j$ using Eqs. (5a) and (5b).

5) Use the $(j = 0)$ level of Eq. (5b) to calculate $x_0$.

Per grid point (i.e., for $j = 1 \to J$), the following operation counts and storage needs are observed: with $\mu = \ell + n - m + k$ $(= \ell + m' + k')$ and $\mu' = \ell + m + k'$ $(= \ell + n - m' + k)$. Steps 2a, 2c, and 4, respectively, involve $(n + k')m\mu$, $k'(n - m)\mu$, and $n(\mu - \ell)\ell$ multiplications (mul) and a similar number of additions (add); totaling $n(\mu\mu' - \ell^2)$ (mul, add); whereas step 2b involves $\mu$ solutions (sol) of $n \times n$ matrix equation. Only, the $\bar{\ }$, $\tilde{\ }$, and $\hat{\ }$ components of $x_j$ and $y_{j-1}$ need to be stored, requiring $n\mu$ locations (loc).

The method defines a decomposition of the coefficients matrix $\Gamma$ of the form $\Gamma = LU$, so that the solution procedure can be divided into two sweeps. In the forward sweep, we solve, for

$$\bar{z} = [(\bar{x}_0)\ (\bar{y}_o\ \bar{x}_1) \cdots (\bar{y}_{j-1}\ \bar{x}_j) \cdots (\bar{y}_{J-1}\ \bar{x}_J)\ (\bar{y}_J\ \bar{q})]^T$$

the equation $L\bar{z} = g$, where $L$ is the matrix

$$\begin{bmatrix}
C_{20} & & & & & & & & \\
A_{11} & \tilde{B}_{11} & C_{11} & & & & & & \\
A_{21} & \tilde{B}_{21} & C_{21} & & & & & & \\
 & & A_{1j} & \tilde{B}_{1j} & C_{1j} & & & & \\
 & & A_{2j} & \tilde{B}_{2j} & C_{2j} & & & & \\
 & & & & A_{1J} & \tilde{B}_{1J} & C_{1J} & & \\
 & & & & A_{2J} & \tilde{B}_{2J} & C_{2J} & & \\
 & & & & & & A_{3J+1} & \tilde{B}_{3J+1} & \hat{E}_{3J+1} \\
A_{41} & \tilde{B}_{41} & A_{4j} & \tilde{B}_{4j} & A_{4J} & \tilde{B}_{4J} & A_{4J+1} & \tilde{B}_{4J+1} & \hat{E}_{4J+1}
\end{bmatrix}$$

In the backward sweep, we solve, for $z$, the equation $Uz = \bar{z}$, where $U$ is the unit upper triangular matrix

$$\begin{bmatrix}
I_m & \tilde{X}_0 & & & & & & & \hat{X}_0 \\
 & I_{n-m} & & \tilde{Y}_0 & & & & & \hat{Y}_0 \\
 & & I_m & \tilde{X}_1 & & & & & \hat{X}_1 \\
 & & & I_{n-m} & & \tilde{Y}_{j-1} & & & \hat{Y}_{j-1} \\
 & & & & I_m & \tilde{X}_j & & & \hat{X}_j \\
 & & & & & I_{n-m} & & \tilde{Y}_{J-1} & \hat{Y}_{J-1} \\
 & & & & & & I_m & \tilde{X}_J & \hat{X}_J \\
 & & & & & & & I_{n-m} & \\
 & & & & & & & & I_k
\end{bmatrix}$$

with the subscript of the identity matrix $I$ defining its order.

The procedure for the direct boundary-layer problem follows as the special case when $k = 0$. To obtain its description, drop all mention of $q$, $E$, and symbols marked with $\hat{\ }$, and then modify Eqs. (1–10) and the matrices $L$ and $U$, accordingly. In particular, Eqs. (5a) and (5b), which define the algorithm, become

$$y_{j-1} = \bar{y}_{j-1} - \tilde{Y}_{j-1}y_j \qquad j = 1 \to J \tag{11a}$$

$$x_j = \bar{x}_j - \tilde{X}_j y_j \qquad j = 0 \to J \tag{11b}$$

The method, in this case, is a reorganization of Wornom's method[4] that leads to a nonsingular $C_{20}$ and, thus, avoids the special treatment adopted by Wornom to start his solution process.

## Conclusion and Assessment

The algorithm introduces a non-bordering strategy that makes solution methods for the direct problem applicable to the inverse problem, as is. A term involving the subvector $q$ of line unknowns is appended to the recurrence relations that define the method. [Compare Eqs.(5) with Eqs. (11).] In contrast, the bordering strategy reduces the inverse problem to a direct problem with several right-hand sides. Equation (1) is put in the partitioned form

$$\begin{bmatrix} \Gamma_- & E_- \\ \Gamma_5 & E_5 \end{bmatrix} \begin{bmatrix} z_- \\ q \end{bmatrix} = \begin{bmatrix} g_- \\ g_5 \end{bmatrix}$$

where the subscript 5 denotes $k$-row submartrices. Solution by bordering amounts to solving the direct problem $\Gamma_-\ [\mathfrak{Z} \mid \mathfrak{z}] = [E_- \mid g_-]\mathfrak{Z}$ and $\mathfrak{z}$, then solving $(E_5 - \Gamma_5\mathfrak{Z})\ q = (g_5 - \Gamma_5\mathfrak{z})$ for $q$, and, finally, evaluating $z_-$ from $z_- = \mathfrak{z} - \mathfrak{Z}q$. (An extra LOOP is needed for this evaluation, unless $k' = 0$ when it can be averted.)

**Table 1 Storage needs and operation counts**

| Algorithm: | BTDMIA | | Present | |
|---|---|---|---|---|
| Bordering: | Without | With | Without | With |
| (loc) | $n(\mu' + n)$ | When $k > k'$ add: | $n\mu$ | When $k > k'$ add: |
| (mul, add) | $n(\mu\mu' - \ell^2)$ | $n(k-k')(n-m)$ | $n(\mu\mu' - \ell^2)$ | $n(k-k')(n-m)$ |
| (sol) | $\mu$ | $k-k'$ | $\mu$ | |
| LOOPs | 2 | 3 | 2 | 3 |

To assess the present algorithm as compared with the BTDMIA, we give, in Table 1, their storage needs and operation counts per grid point, when both are applied, without and with bordering, to the inverse boundary-layer problem described earlier. For fair comparison when $m > m'$, the two methods should operate in opposite directions. The forward sweep of the present algorithm is to be carried out starting from the side with $m$ conditions, whereas its counterpart of the most efficient form[5] of the BTDMIA is to be carried out starting from the other side.

The conclusion is twofold: non-bordering is superior to bordering. The present algorithm is more efficient than the BTDMIA.

## References


[1]Bradshaw, P., Cebeci, T., and Whitelaw, J. H., *Engineering Calculation Methods for Turbulent Flows*, Academic Press, New York, 1981.

[2]Chen, H. H., and Cebeci, T., "Bordering Algorithm for Solution of Boundary-Layer Equations in Inverse Mode," *AIAA Journal*, Vol. 29, No. 12, 1991, pp. 2257–2259.

[3]Cebeci, T., Keller, H. B., and Williams, P. G., "Separating Boundary-Layer Flow Calculations," *Journal of Computational Physics*, Vol. 31, 1979, pp. 363–378.

[4]Wornom, S. F., "A Fourth-Order Box Method for Solving the Boundary-Layer Equations," NASA TM X-74000, Jan. 1977.

[5]Keller, H. B., "Accurate Difference Methods for Nonlinear Two-Point Boundary Value Problems," *SIAM Journal of Numerical Analysis*, Vol. 11, No. 2, 1974, pp. 305–320.


# Derivation of a Modified Hybrid Approximation


Thomas M. Harms,* Theodor W. von Backström,† J. Prieur du Plessis,‡ and Louwrens M. Toerien§

*University of Stellenbosch, Stellenbosch 7600, South Africa*


## I. Introduction

CONTROL volume interface interpolation schemes can be classified as to whether they are local grid- or streamline orientated and whether they are algebraically or physically based. The hybrid scheme described by Spalding[1] is an example of a gridline-orientated physically based scheme. The exponential coefficient function resulting from the analytic solution of a one-dimensional homogeneous convection-diffusion equation, i.e.,

$$A(Pe) = Pe/(e^{Pe} - 1) \tag{1}$$

with $Pe$ the Péclet number, is approximated through its asymptotes, resulting in

$$A(Pe) \cong \max(-Pe,\ 1.0 - Pe/2.0,\ 0.0) \tag{2}$$

Both Eqs. (1) and (2) result in upwinding at high Péclet numbers, which is inadequate in terms of accuracy in multidimensional situations due to the inherent artificial diffusion as pointed out by Pulliam.[2] Algebraic grid-aligned interpolation addresses this problem through higher order interpolation schemes, e.g., central differencing used by Peric',[3] quadratic upwinding described by Leonard and Mokhtari,[4] or other variations employed by Shyy et al.[5] With regard to physically based interpolation schemes the remedy has been the employment of the solution of a nonhomogeneous convection-diffusion equation, i.e.,

$$\frac{\partial}{\partial \xi}\left(\rho u \phi - \Gamma \frac{\partial \phi}{\partial \xi}\right) = S \tag{3}$$

in both flow-aligned and grid-aligned schemes. Examples can be found in the work of Raithby[6] and in particular control volume finite element schemes such as described by Baliga and Patankar,[7] Prakash,[8] and Schneider and Raw.[9] Furthermore, the function $S$ not only provides the opportunity to account for multidimensional effects but also provides a velocity-pressure coupling allowing the use of nonstaggered grids. This was previously shown by Prakash[8] for flow-aligned interpolation and Thiart[10] for grid-aligned interpolation. It is the physically based analog to the methods of Baliga and Patankar[7] or Rhie and Chow,[11] which are based on central difference velocity interpolation.

Harms et al.[12] previously showed on the basis of orthogonal grids that in the context of Eq. (3) streamline-orientated interpolation with parabolic interface flux integration represents an unnecessary complication with regard to accuracy. Although physical interpolation is realistically attractive, it is at best second order when based on integration that assumes constant properties over surfaces and volumes. It is, on the other hand, computationally expensive and associated with instability as reported by Huang et al.[13] In this Note we wish to address the issue of computational expense by examining two alternatives to obtain and deal with coefficient and other exponential weighting functions arising out of the application of Eq. (3).

## II. Derivation

Schneider and Raw[9] proposed to obtain an interface value in the following manner: $\rho u$ and $\Gamma$ are assumed constant in the space $\xi$ (0 to $L$) between two nodes and known from a previous iteration. The first derivative on the left of Eq. (3) is then discretized as an upwind difference, the second derivative as a central difference, and the source term $S$ (at first $S_0 = S_L = S$) in any appropriate manner (e.g., a pressure gradient in the interpolation direction would follow from the adjacent nodal pressure difference and cross fluxes and other source terms from arithmetic averages of terms discretized at the two adjacent nodes[12]). For a noncentralized interface ($a = \xi/L$) and positive or negative interface velocity, the result is

$$\rho u \frac{\partial \phi}{\partial \xi} = \frac{\Gamma}{L}\left[\max(0, Pe)\frac{\phi_{aL} - \phi_0}{aL} - \max(-Pe, 0)\frac{\phi_L - \phi_{aL}}{(1-a)L}\right] \tag{4}$$

$$\Gamma \frac{\partial^2 u}{\partial \xi^2} = \Gamma \frac{a\phi_L - \phi_{aL} + (1-a)\phi_0}{a(1-a)(L^2/2)} \tag{5}$$

for $0 < a < 1$ and $Pe = \rho u_{aL} L/\Gamma$. Substituting Eqs. (4) and (5) into Eq. (3) results in

$$\phi_{aL} = \phi_0 + \frac{a}{k}[\max(-Pe, 0) + 2](\phi_L - \phi_0) + \frac{L^2}{k\Gamma}a(1-a)S \tag{6}$$

Received Sept. 27, 1993; revision received Dec. 20, 1993; accepted for publication Feb. 22, 1994. 


*Graduate Student, Department of Mechanical Engineering.

†Professor, Department of Mechanical Engineering.

‡Associate Professor, Department of Applied Mathematics.

§Graduate Student, Department of Applied Mathematics.